PipeMeta 0.44 (beta) Documentation

PipeMeta is a perl script designed to assist in running an annotation pipeline for 454 transcriptome projects. The average transcriptome annotation project generates several files. This includes the raw read (primary EST file) to start the trimming and quality control process before assembly. Typical post-assembly output includes an assembly alignment export file containing all the information required to recreate the assembly (Pipemeta uses the ace file format), the assembled consensus sequences (i.e. Contig file), the remaining non-assembled sequences (i.e. Singleton file), and all the reads incorporated into the assembly (i.e. assembly EST file). In addition, the Lasergene assembler, Seqman pro, generates an assembly report file containing read assignments and quality trim information and a statistics file containing various useful data on the Contigs, although all these data can be gleaned from the ace file alone if necessary. All of these files should be placed in a single 'working' folder (and a project created for this folder) in order to allow Pipemeta to access them during the pipeline process. The pipeline itself consists of several perl scripts optimized for Windows platforms (tested on Vista), although many of the individual scripts should run successfully on other platforms. PipeMeta runs from the command prompt window (e.g. C:\PipeMetaFolder>PipeMeta) and uses a very basic menu system to greatly simplify the pipeline process. Pipeline progress and resulting file names are tracked under a simple file system as projects. The various PipeMeta menus are for the most part ordered to reflect the natural progression of the typical pipeline project, while still allowing for some flexibility in the annotation process. The following documentation follows the basic outline of PipeMeta's menu system.

User Notes: Some important thoughts to keep in mind.

- Pipemeta was intended as a convenient shortcut or alternative means to run a *de novo* transcriptome pipeline. It saves time (compared to running the pipeline scripts manually) and assists in streamlining and organizing the process. However, the various pipeline scripts are capable of being utilized independently and have their own help features and limited documentation that can be accessed by entering help option after the script name (e.g. C:>Script.exe –h). In some cases (a more customized project), running the scripts individually may be the better option.

- The design and menu layout of Pipemeta was intended to allow for some flexibility in the pipeline process. For example, using the command line legacy Blast on the typical workstation to run a Uniprot analysis can easily take several days or even a couple of weeks. However, Pipemeta allows any tab-delimited Blasts results to be incorporated into the project, so one could run the Blast on a cluster and add the results into the next pipeline step if desired.

- Pipemeta can have difficulties with non-standard file and folder names. It is highly recommended that PipeMeta and all working folders be located in directories without spaces, dashes, or non-extension periods in the names (e.g. Windows XP locates the desktop in 'Documents and settings' directory, which will cause errors).

To avoid potential errors, all PipeMeta associated files and folder/directories should only contain alpha-numeric characters, underscores, and one period for the file extension (e.g. File1_date.txt).

- PipeMeta is a work in progress, please feel free to contact the author with any comments, questions, issues, or bugs. Known issues at the moment:
    - Varsplic does not compile correctly and thus is not available for the executable PipeMeta package. This is a problem (as far as I can tell) with the Swissknife perl package it uses, and so is not an issue I can personally correct. Use one of the two other Uniprot formatting options instead, or use the perl source PipeMeta package (available by request).

Installation: Pipemeta does not require installation itself, simply place it and all related scripts into a folder/directory with no non-alphanumeric characters other than underscores and run it from the command line window (e.g. C:>PipeMeta/PipeMeta.exe). However, several of PipeMeta's options require pre-installed software to be present, most importantly BioEdit and MySQL. Luckily, these software are open-source and freely available for download from their webpages (see below). Also, running PipeMeta from the perl source code requires perl to be present and several additional perl modules/packages to be installed (contact author for details if required).

Running PipeMeta: PipeMeta should be run from within a Windows command line shell (e.g. C:>PipeMeta\PipeMeta.exe) for ease of use. It can be opened directly from Windows, however (for whatever reason), opening it in this fashion prevents cutting and pasting to and from the command prompt window, thus costing the user a major time saving device. Cutting and pasting to the command prompt window is especially handy for copying folder paths from a Windows window when PipeMeta prompts for them. Simply copy the folder path from the Windows window of the desired folder and right click on the command prompt window to bring up the cut and paste menu. Having a shortcut to the command prompt window on your desktop is equally handy. You can even go into the shortcuts properties (right click on it) and change the 'Start In' directory to the Pipemeta folder path.

Main Menu:

1) *Project Menu*: Manage projects and change the working directory of a project. Projects are the primary means of keeping track of related annotation files and pipeline progress. The working directory is where PipeMeta looks for project files and directs its output. All project files should be kept in this folder. Project information is stored to a text file (e.g. PipeMetaProject_[projectname].txt) in the PipeMeta directory. MySQL tables are organized by project name as well as MySQL database/schema for uploading and downloading purposes.
    - *View current project name and working directory*: enter 'y' or 'yes' to make changes.
    - *New Project*: create a new project. Select name and working directory for the

project.

- *Open project*: open/switch to an existing project.
- *Delete Project*: delete a project.
- *Change/set working directory*: set the working directory for the currently active project.

2) *Scan Files menu*: Generate basic statistics for sequence and quality score files and view Contig assembly alignments from ace files.

    - Scan Sequence Files: Scan sequence and quality score files for basic statistics.
        - *Scan Sequence Files*: Statistics generated include number of nucleotides, average sequence length, standard deviation, and coefficient of variation.
        - *Scan Quality Score Files*: Statistics generated include number of quality scores, average sequence quality score, standard deviation, and coefficient of variation.
    - View Contig Assembly (Bioedit): searches ace file and displays an assembly alignment for a single Contig using Bioedit. Text search is case-sensitive. Bioedit should be installed in the default directory (i.e. C:/bioedit/). To save the alignment to file, copy sequences to clipboard (Fasta format) and open a new bioedit instance. Import the sequences from clipboard to the new instance and save to file. Bioedit is available at: http://www.mbio.ncsu.edu/BioEdit/bioedit.html

3) *Pre Assembly Menu*: Run pipeline scripts to combine files, screen for SMART primer, modify/trim sequences and quality scores, and edit titles in order to prepare them for assembly. The trim/edit menu can often be used for post assembly functions as well, however.

    - Combine Sequence Files (Fasta format): Combine files from working directory or specify a source directory. Choose from a selection of available Fasta files (i.e. file1.fas). If none are available, Pipemeta will prompt for a file extension to select from. Select two or more files to combine and then specify output file name.
    - Create or Add to Smart Primer File: create a text file containing one or more tab-delimited lines of Smart primer core sequence and antisense reverse compliment sequence. Also allows additional line of sequence to be added if smart primer file already exists in the working directory. Hit enter at the prompt for primer sequence to create a file from default sequence (i.e. defualt 5' core SMART Oligo II and CDS: AAGCAGTGGTATCAACGCAGAGTACGC).

- Screen 454 EST File (Fasta Format) for Smart Primer: Screens a file for SMART primer by running the SmartScreener pipeline. 454 ESTs should be in a FASTA format text file in the working directory. Accept current project EST file or reject to specify another input file name. Pipemeta will then verify the input and output file name choices. Enter yes to begin pipeline or no to specify another output file name.

- Modify 454 EST Quality Score File (Fasta Format): Pipeline to filter sequences and quality score files (e.g. .fas and .qual) for screened primer (i.e. n's generated by SmartScreener) and low quality sequence using various sequence quality filters. Est sequences and quality scores are specified and then verified. Enter yes to begin pipeline or no to specify another output file name. An additional menu then appears to specify pipeline parameters.

    - Convert primer quality scores to zeroes: quality scores corresponding to screened primer are converted to zeroes. No trimming or quality filtering occurs.

    - Trim primer nucleotides and quality scores, no filters: quality scores and screened primer nucleotides are trimmed off the reads and quality score sequences. No quality filtering occurs.

    - Trim primer nucleotides and quality scores, use filters: quality scores and screened primer nucleotides are trimmed off. Sequence quality filters are active and Pipemeta will prompt for filter parameters. Filters include maximum degenerate nucleotides per sequence, minimum average quality score per sequence, maximum sequence length, and minimum sequence length. Some recommended filter settings are 1, 20, AverageLength+1*SD, and AverageLength-1*SD respectively, where SD is the standard deviation of the average sequence length of all reads.

- Trim/Edit Sequences: Pipeline interface to scripts with various sequence trimming and title editing capabilities.

    - Trim Contig/Singleton/Reference sequence: If trimming a reference set, select the desired set from menu or cancel to specify the file name. Accept Contigs/Singletons from project or specify Contig/Singleton file input to trim/filter sequences. Enter yes at verification prompt to begin pipeline or no to specify another output file name.

        - *Trim by length and remove n's, no splitting*: removes n's from sequences and then filters by a specified minimum length. Filtered sequences are dropped from output. Sequences are never split. Intended for sequences with screened primer (i.e. 454 Singletons).

- *Trim by length and remove n's, with splitting*: removes n's from sequences and then filters by a specified minimum length. Filtered sequences are dropped from output. Sequences containing an internal string of screened nucleotides (i.e. n's) are split into two new sequences. Multiple internal strings produce an error. Intended for sequences with screened vector (i.e. Sanger Contigs).

- Edit PrimaryESTs/Contig/Reference: If editing a reference set, select the desired set from menu or cancel to specify the file name. Accept Contigs/Singletons file from project or specify Contig/Singleton file input to edit sequence titles. Enter yes at verification prompt to begin pipeline or no to specify another output file name.

    - *Split sequence titles using delimiting characters and select resulting chunks*: specify a character string (e.g. STRING) that will be used to delimit the sequence title into 2 or more sub-strings (i.e. chunks) (e.g. Subtitle1STRINGSubtitle2 -> Subtitle1 Subtitle2), then select the desired chunks to retain in the title (i.e. 1 = Subtitle1, 2 = Subtitle2). Some less common delimiting characters are illegal and result in an error. Next Pipemeta prompts for the option to Chomp the resulting chunks in order to remove the last character. This is useful for removing some illegal characters that would otherwise be irremovable. Next is an option to specify a character string to insert in between the reassembled title chunks. Hit enter to insert none.

    - *Append characters to titles*: specify a character string to append to sequence titles.

- Edit Ace file Contig titles: edit Contig titles within an Ace file by inserting a character string between the word 'Contig' and the Contig ID number (e.g. Contig100 -> Contig_100). Intended to correct inconsistently titled Seqman Pro output.

4) *Post Assembly Menu*: Runs the annotation and SNP finding pipelines through a series of menus in roughly logical order, from blast comparisons to the Uniprot database, to file preparation and upload to a mySQL server, SNP discovery, and tab-delimited annotation table creation, all run locally.

- Blast Menu: Menu options for downloading, installing, and running Blasts locally (as opposed to over the internet). Typically, the annotation process begins with Blast alignment of assembled sequences to a major annotated database. Pipemeta is designed to use the Uniprot protein knowledgebase for annotation purposes (http://www.uniprot.org/). Uniprot consists of two sections: Swiss-Prot, which is manually annotated and reviewed, and TrEMBL, which is automatically annotated and is **not** reviewed. Uniprot has the advantages of containing a curated, all-protein database, which allows for higher quality annotation as well as frame and codon position calculation with some degree of certainty. Also, being a protein database makes for a much smaller storage solution than the equivalent nucleotide database, and thus a faster alignment analysis. In addition, use of the (optional) RefSeq clustered sequence file (where identical Uniprot entries are combined into a cluster while the best available sequence and annotation is kept to represent the cluster) for Blast analyses improves both the quality of the annotations and the runtime even further. A disadvantage of using Uniprot is that it lacks representation of UTR regions, which constitutes a significant portion of most transcriptome sequencing projects. However, UTR is known to be less conserved between species than ORF regions, making blast analyses less useful for identifying these regions. In addition, UTR is less commonly present, even in large nucleotide databases (e.g. NCBI's genbank).

    - Install local NCBI Blast algorithm: Download and install the latest Blast binary version from NCBI (ftp://ftp.ncbi.nlm.nih.gov/blast/executables/LATEST/). Choose between 32-bit and 64-bit processor versions, and between Legacy Blast and Blast Plus. In order for the binary to work properly, be sure to copy the NCBI configuration file to the Windows folder when prompted (for Legacy Blast).
    - Setup local Uniprot Database: Download and setup a local version of Uniprot knowledgebase for Blast alignments and indexed annotation retrieval. Options include the complete Uniprot indexed database with varsplic, non-varsplic, or UniRef100 Blast component, or specific taxonomically divided portions of Uniprot (e.g. plant). Projects can only have one active Uniprot database at a time, although more than one can be downloaded and set up. Pipemeta can store the locations of the resulting databases by user specified name for later Blast alignment and annotation retrieval. The setup process is divided into three independent steps.

        Note: There are strengths and weaknesses to using either Uniprot or Uniref for annotation purposes. Uniref combines identical (but different size/position) proteins into single entries and provides the best possible annotations (i.e. avoids the annoying and uninformative 'predicted protein' where possible), while Uniprot allows for more sensitive Blast alignments (especially with Varsplic option) and taxonomy matches.

        - *Download Uniprot database*: Pipemeta checks what Uniprot DBs are available online and then downloads and unarchives the selection.

- *Create indexed Uniprot annotation database*: creates an indexed annotation DB from a Uniprot flatfile for fast annotation retrieval.

- *Create Uniprot Blast database*: creates a Blast DB from either the Uniprot flatfile (Varsplic or DatParser) or the UniRef100 fasta file (Uniprot Complete only) using the NCBI FormatDB tool.

- Setup local Blast database: Creates a local Blast database from a sequence file using NCBI FormatDB tool (Legacy Blast only) and stores the location in the PipeMetaSettings text file under a user specified name.

- Add/Remove local Uniprot/Blast database: Add or remove a local Blastable sequence database that *has already been formatted*.

    - Add Blast sequence DB: Add a ***formatted*** local Blast/Uniprot database to PipeMetaSettings.txt file, making it available on the menu list of Blastable DBs to choose from. NOTE: The formatted DB must follow *all* Pipemeta naming conventions for custom Blast DBs or errors may occur. In particular, the file containing the fasta formatted sequences must use the '.fas' extension and the DB folder(s) should have the form of either 'BLAST_[DB_Name]' or 'INDEX_[DB_NAME], depending on whether it's a Blast folder or an indexed Uniprot annotation folder.

    - Remove Blast sequence DB: Remove a local Blast/Uniprot DB from the PipeMetaSettings.txt file and delete associated files and folders.

- Blast sequences (Fasta format): Perform a sequence alignment using NCBI's Blast alignment tool on a Fasta formatted sequence file. The file can contain multiple sequences. At the prompt, Enter 'yes' to perform Blast analysis on project contigs or singletons for annotation purposes, or enter 'no' to select another annotation database or to perform a non-annotation sequence alignment. Annotation Blasts can be performed using either Legacy Blast or Blast Plus, depending on setup (see above). For large projects, Blast Plus is recommended, as it is much faster. If there is no Uniprot/UniRef100 DB already setup or if 'no' is entered, a menu of all available Blast databases will appear. Select a DB or select 'Other' to enter a new DB name and location. After the Blast DB has successfully been selected, Pipemeta will again prompt to make the selection the default Uniprot or UniRef100 DB. NOTE: only actual Uniprot/Uniref100 DBs, with indexed annotation component, should be selected for pipeline annotation. Next, select the query type of the sequence file to align. Select

'Non-Project' to perform a non-project related Blast alignment (i.e. non-annotation pipeline; e.g. to blast a non-project sequence against project contigs). Next, select the Blast flavor to use, number of processors to utilize, and whether or not to use low-complexity filters (typically always used except when aligning same species sequences). Non-project Blasts additionally prompt for preferred output format (standard output with alignments shown or tab-delimited) and maximum number of output subjects per query (typically 5 for other pipeline blasts). Finally, Pipemeta prompts for verification of the blast inputs and outputs. Type 'yes' to begin the Blast and 'no' to change the output file name or to cancel Blast.

- Annotation Menu: The next step in the pipeline process, the annotation menu contains options for preparing files for upload to a relational server/client database (i.e. MySQL) and for retrieving annotation from the indexed Uniprot flatfile.
    - Prep sequence files: Prepare project sequence files (Fasta format) for upload to MySQL DB.
        - Parse primary EST files: Combines information from the EST sequence file and EST quality score file used in the assembly (if there were more than one each they can be combined using the Combine Files option, see above) to create a tab-delimited text file with a title line.
        - Parse Singleton EST file: Prepare Singleton file for upoad by converting to tab-delimited format and adding a title line.
        - Parse Reference Sets: Opens an additional menu of available sequence reference files. Select 'Other' to type in the file name and identifier string (e.g. species name) of another reference sequence file. Prepares the sequences for upload as above.
        - Parse Seqman assembly report and assembly EST files: Combines information from the Seqman Pro 8.0 assembly report and the assembly EST sequence file into a tab-delimited upload file, both output files from Lasergene Seqman pro assembler program. Edited contig assembly alignments are not adequately documented in the Seqman pro assembly report, however, so if extensive (i.e. new contigs created) editing is performed on the assembly within Seqman pro, the assembly EST upload file should be parsed from the ace (phred/phrap) assembly output file (see below).

- Parse Seqman assembly stats and contig files: Combines information from the Seqman pro 8.0 assembly statistics and the Contig files into a tab-delimited upload file both of which are output files from the Lasergene Seqman pro assembler program. Provides the most accurate statistics for average EST depth per Contig and should be used in preference to the ace file parsing option when available (see below).

- Parse ace file for assembly ESTs or Contigs: Parses the assembly output file (ace format, e.g. standard phred/phrap output) in order to create the assembly EST and/or Contig upload files. For non-Seqman pro assemblies, this is the only available option, although the average EST depth statistic is not as accurate when parsed from the ace file (the parser does not create the sequence alignments necessary for precise statistics). This option is also necessary in order to generate the assembly EST upload file for Seqman pro assemblies that required extensive editing (i.e. new Contigs created, see above), however, in this case the ace file requires additional parsing in order to correct Contig ID errors generated in the ace file by Seqman pro (soon to be made available). The Seqman pro stats file is always accurate and should be used when available.

- Indexed Uniprot unique annotation retrieval: Retrieve a set of unique uniprot annotations from an indexed unprot flatfile database by parsing a list of Uniprot IDs from a Blast output. Uniprot IDs are parsed out into a list and then duplicate IDs are discarded in order to generate only unique annotation entries for any given Blast file. Pipemeta will suggest a Uniprot DB to use if a default has been set. Enter yes to use the default or no to select another indexed Uniprot database from a menu. NOTE: only an *indexed Uniprot* database should be selected, although all available databases will be displayed. Once the indexed database has been chosen, a menu of available project Blasts will display. Choose a project Blast file (e.g. Contigs) or select 'Non-Project Blast' to incorporate a Blast file not yet saved to the project. Choosing reference set Blast file will bring up a new menu of available reference sequence (vs. Uniprot) Blast files.

- Prep annotation files: Prepare project Blast and Uniprot annotation files for upload to a MySQL DB.

    - Prep Uniprot annotation/Blast files: Prepare available (i.e. saved to a project) Uniprot annotation for upload. Select Contig, Singleton, or a reference set annotation file to prepare if available,

or select 'Use other annotation file' to type in a file name manually and then select the query type. Creates two UPL files ready for upload to MySQL tables.

- Prep Other Blast file: Prepare available (i.e. saved to a project) non-Uniprot related Blasts for upload. Non-Uniprot Blasts are needed to create multiple-subject annotation tables. An example would be a Contig table that included Blast alignments to both Uniprot and a reference sequence set. In this case two separate Blasts would be required: Contigs vs. Uniprot and Contigs vs. the reference set. First step is to select the query type of the Blast file to be parsed, which brings up the project files available for that type. Choose 'Other' to type in a file name first and assign the query and subject type.

- Prep KEGG file: Parse and prepare a KEGG hierarchy file for upload. KEGG (Kyoto Encyclopedia of Genes and Genomes; http://www.genome.jp/kegg/) provides another level of annotation that can be included in the Pipemeta pipeline as an optional table in the MySQL database. Only reference sets that have Uniprot annotation can be uploaded to the MySQL DB. In order to generate KEGG annotation for the pipeline, the first step is to submit a file containing both the Contigs and Singletons (or a sequence reference set) to the KEGG automated annotation server (i.e. KAAS; http://www.genome.jp/tools/kaas/). Once the automatic server pipeline is finished (a process that can take several days), an archived folder containing the KEGG hierarchy files can be downloaded from KAAS (usually named hier.tar.gz). Unarchive the folder and copy the top file (usually named q00001.KEG) into the project working directory. Rename the KEGG file if there is already KEGG annotation for other sequence sets in the directory. The file contains the primary KEGG IDs and hierarchy level definitions for the submitted sequences. Next, enter the name of the KEGG file when prompted and specify whether the KEGG file is for Contigs and Singletons or for a reference set.

- Database Menu: The database menu has options for connecting, uploading, and downloading sequence data from a MySQL database. MySQL is an open source, freely available relational database (found at: http://dev.mysql.com/downloads/mysql/) that provides a powerful means of storing and retrieving transcriptome data from either a local or server/client setup. MySQL should be downloaded, installed, and configured prior to the upload step of the pipeline. The MySQL database is where all sequence data and annotation generated by Pipemeta is ultimately intended to be stored. Once the upload process is complete, all project directory files are no longer needed (with the exception of SNP related files), although they should be archived and stored as a

backup source.

- Setup MySQL: Manage connections to a MySQL server. Allows connection parameters to be named and stored for convenience, except for passwords, which must be entered for each Pipemeta instance. Also allows specific databases (i.e. schema) to be created. Think of databases/schema as sub-databases within the larger, meta-database MySQL instance. It is recommended that individual projects be stored on separate database/schema, although this is not required.
    - New Connection: Create and store a new MySQL connection to PipeMetaSettings.txt. Only alpha-numeric characters and underscores are supported. The MySQL database should already be created prior to uploading data, but can be entered here first safely. The parameters to be specified are:
        - *Connection Name*: Name of the connection for storage purposes. Useful names specify the user name and location of the database (e.g. John_LabServer).
        - *User Name*: MySQL user name.
        - *Database Name*: MySQL database/schema name. Can be created from Pipemeta (see below).
        - *Host Address*: IP address or domain name of the MySQL server. Use 'localhost' if connecting from the server machine.
        - *Port Number*: The default MySQL port is 3306, although this can vary.
        - *Password*: MySQL user password. Passwords are not stored by Pipemeta and must be entered for every Pipemeta instance.
    - Open Connection: Choose from a list of previously stored MySQL connections. The user password cannot be stored and must be reentered.
    - Remove Connection: Allows unwanted stored MySQL connections to be deleted from storage.
    - Create User: Create new user within a MySQL instance. Specify user name, password, and privileges. *Global* privileges apply to all database/schemas, while Project privileges apply only to a single

database/schema. *Administrators* can upload and download data, while *Members* can only download data. **Note: Only the root user can create new users/assign privileges.**

- Create Database: Create a new database/schema within a MySQL instance. MySQL schemas can be thought of as sub-databases within a MySQL instance. It is recommended that all projects be relegated to their own, separate database/schemas with the same name as the project (although not required).

- Upload project pipeline: Upload project files (UPL files) to a MySQL database. Requires that MySQL be installed and configured and that the connection be set beforehand or an error will occur (see above). Some project file MySQL tables have dependencies (due to the normalization/indexing process) and so Pipemeta will attempt to upload project files sequentially in the proper order. The dependency hierarchy for file upload is: PRIMARYEST_CONTIG_REFERENCE>SINGLETON>KEGG_ASSEMBLYEST_ANNOTATION>BLAST.

- Retrieve project data: Download data from a MySQL database using a variety of means and methods.

  - Search Database: Search a MySQL database for specific project information. Methods:

    - *Keyword search*: search the project database for Contig, Singleton, or Reference entries containing one or more keywords (comma-delimited). One or more fields to be searched can then be specified (comma-delimited, e.g. descriptions, GO terms, KEGGS). Results can be filtered by a minimum bitscore if desired. If KEGG annotation was created and uploaded, it can be included with the results of the search, and it must be included if KEGG was specified as a searchable field. Results will display in the command console if only one search was performed, however, they can be saved to file once the result display is ended.

    - *Find Sequence Annotation*: search the project database for Contig, Singleton, or Reference entries by sequence name. Scans the sequence name field for exact matches and returns all corresponding entries. More than one sequence name can be searched by listing separated by commas. Results can be filtered by minimum bitscore if

desired. If KEGG annotation was created and uploaded, it can be included with the results of the search. Results will display in the command console if only one search was performed, however, they can be saved to file once the result display is ended.

- *Find Contig Assembly ESTs*: retrieve all ESTs for one or more project Contigs by listing the Contig names separated by commas. Saves information to a file.

- *Find Primary EST Info*: search for EST sequence information by listing EST names separated by commas. Saves information to a file.

- *Find Reference Info*: search for Reference Set Sequence information by choosing a Reference Set from a list and then listing Reference Set sequence names separated by commas. Saves information to a file.

- Create full annotation tables: create various summary tables containing sequence and annotation data, including a combined Blast table suitable for importing into SNPHunter. Methods:

  - *All Uniprot*: Create a summary table containing all available Uniprot hits (up to five subjects) for Contigs, Singletons, or References with Uniprot annotation. This is basically a dump of all Uniprot annotation for a group of sequences into a tab-delimited text file. Option to filter output by minimum bitscore if desired.

  - *Combine Top-Hit Annotations*: Create a summary table containing the top Blast hits for one or more Blast analyses (i.e. subject fields) stored to the project for Contigs, Singletons, or References. Useful for creating a multi-Blast subject table for the Aceparser/SNPHunter pipelines. This query retrieves only the top hit for each related project blast (i.e. associated with the chosen query sequences: Contigs, Singletons, or a references) and links them into a single tab-delimited table containing all the basic statistics and annotations for those sequences.

- SNP File Menu: Options for parsing and analyzing assembly program output (i.e. Ace files or phred/phrap output) and combining it with quality score and annotation data for SNP discovery and analysis.

- Create ORFs: Runs a script for finding ORFs from sequence files and creates a tab-delimited file containing the largest in all six frames (two or more sequences can be returned if there is a tie), along with useful positional, frame, and size information. Useful for frame and codon position detection when running SNPHunter.

- Create annotated assembly file: Pipeline for running AceParser script. Parses an ace file and converts it into an annotated, tab-delimited assembly alignment file (i.e. ass file). Also parses and formats an associated quality score file for inclusion in the SNPHunter pipeline (see below) if available. For smaller projects with quality scores, whole assembly mode provides the fastest pipeline run time. For very large projects with quality scores, memory requirements can be prohibitive, and split assembly mode is the best choice. Most average systems can handle about 200,000 quality score reads in memory at once (depending on the read length), so a project with 600,000 quality reads would require at least a split number of three. The annotation is imbedded in the assembly file (unlike the quality scores) and can be derived from one of two formats: A typical tab-delimited NCBI Blast output file with a single subject column or a multi-subject columned Blast/annotation table created using the '*Create full annotation tables-> Combine top-hit annotations*' option in the Database Menu. In order for the table to be parsed correctly, add the Uniprot query last (after adding other desired queries), so that the Uniprot Blast data and annotation appears on the far right side of the table. When the AceParser pipeline runs, the BlastColumnFilter script will automatically parse the multi-subject column annotation table to derive the best available Blast information in order to orient and position the SNPs during the SNP search.

- Find SNPs: Pipeline for running SNPHunter to search for high quality SNPs within the assembly. Uses an annotated assembly alignment file (i.e. ass file, see above) and a padded quality score file as primary inputs, as well as three other optional input files. The formatted (i.e. padded) quality score file (created using AceParser, see above) is used to filter out low quality nucleotides in the assembly and for calculating SNP quality statistics. SNPHunter has been optimized to search *de novo* transcriptome assemblies (assemblies without using a reference genome). It uses a series of filters and search parameters to screen out suspect assembly regions and areas of polymorphism not likely to be allelic in nature (e.g. alternative splicing, sequencing/base call errors due to homopolymer runs). SNPHunter's output depends to some degree on the parameters set, but is basically a tab-delimited table containing the SNPs and related statistics that can be opened using most commonly used spread-sheet software (JMP is recommended by this author). In SNPHunter v2.1 and higher, the filters are generated (as a text file) by the AssemblyFilter script and read by

SNPHunter as an optional input. SNPHunter also has the ability to incorporate population information with labeled read names and a simple population tag key file. The third optional file read by SNPHunter is a file containing ORF information (see above in Create ORFs) that is used to help determine SNP codon positions (along with the Blast data).

To run SNPHunter, first determine whether to use whole or split mode. Follow the prompts to choose quality scores, population parsing key, ORF data, and input/output file names. Next, select which filters and parameters to use from the interactive options menu. The default (recommended for most applications) settings are displayed on the menu when it appears. If any of the filters are selected, AssemblyFilter will run prior to SNPHunter. For more information on the filters and parameter settings, see the SNPHunter Overview or Docs.

- Additional pipeline scripts:

Some additional annotation pipeline scripts have been deemed sufficiently useful to be included in the PipeMeta package that cannot (currently) be run from within PipeMeta. As in all my scripts, they contain some documentation viewable using the help option.

E.g. C:>Script1.exe –h

Worth mentioning here as well is a useful feature (also not currently available from within PipeMeta) of the BlastRowFilter script, which is the ability to filter out undesirable Blasts/annotations using a list of terms to be excluded (actually, these terms are given the lowest priority in a sorting algorithm). This feature allows an alternate set of top annotations to be created from the full Blast annotated Blast set (which is easily generated from within PipeMeta using the '*Create Full Annotation Table -> All Uniprot*' menu options).

- *AdapterSearch*: Remove 454 and/or Illumina adapter sequence from raw reads while tagging the corresponding read titles with an alpha-numeric string. Useful for identifying source population of reads after they've been combined into an assembly. This procedure is usually (but obviously not always) accomplished by the sequencing facility, in which case the reads will have already been adapter filtered and sorted into separate files. In this case, the read titles only need to be tagged with an identifying alpha-numeric string, which can be accomplished from within PipeMeta (using primarily the ColumnEdit script).

- *ExpandSnpTable*: Expands the SnpName and AlleleName SNPHunter output fields into multiple separate columns for ease of downstream SNP analyses. Expanded columns are SnpAlleles, MajorAlleleDepth, MinorAllele1Depth, MinorAllele2Depth, and MinorAllele3Depth.

- *SnpSnap*: Cut out contig sequence regions around SNPs using customizable size and quality parameters for use in PCR primer design.

Contact Info: please feel free to contact for questions/comments and for reporting bugs/issues.

Author: J. Cris Vera

Email: jcv128@psu.edu

Phone: 814-863-5927